

SYSTEM FOR SYNCHRONIZING CIRCUITRY IN AN ACCESS NETWORK

10

BACKGROUND

This invention relates to digital computer network technology. More specifically, it relates to methods and apparatus for synchronizing components within the Head End of an access network.

In conventional Data Over Cable Service Interface Specification (DOCSIS) systems, there may be multiple Cable Modem Termination Systems (CMTSs) each including a plurality of physically distinct line cards having appropriate hardware for communicating with cable modems in the network. Each CMTS and each line card is typically assigned to a separate DOCSIS domain, which is a collection of downstream and upstream channels.

Typically, each DOCSIS domain includes a single downstream channel and one or more upstream channels. The downstream channel is used by the CMTS to broadcast data to all cable modems (CMs) within that particular domain. Only the CMTS transmits data on the downstream. In order to allow the cable modems of a particular DOCSIS domain to transmit data to the CMTS, the cable modems share one or more upstream channels within that domain.

Access to the upstream channel is controlled using a time division multiplexing (TDM) approach. Such an implementation requires that the CMTS and all cable modems sharing an upstream channel within a particular domain have a common concept of time so that when the CMTS tells a particular cable modem to transmit data at time T, the cable modem understands what to do. "Time" in this context is tracked using a counter, commonly

5 referred to as a timestamp counter, which, according to conventional implementations is a 32-bit counter that increments by one every clock pulse.

 In conventional configurations, each line card in each CMTS may have its own unique timestamp counter which generates its own local time reference. Thus, each line card in the DOCSIS system operates according to its own local time reference, and is not synchronized
10 with other line cards in the same CMTS or with line cards in other CMTSs.

 Each line card in the DOCSIS system periodically distributes a timestamp value of its local time reference to the respective group of cable modems serviced by that line card. For this reason, a first group of cable modems serviced by a first line card in a first CMTS will not be synchronized with a second group of cable modems serviced by a second line card in a
15 second CMTS.

 The present invention addresses this and other problems associated with the prior art.

BRIEF DESCRIPTION OF THE DRAWINGS

 FIG. 1 is a block diagram of a synchronization system used for an access network.

20 FIG. 2 is a block diagram showing individual Timestamp Synchronization Circuits (TSCs) in the access network.

 FIG. 3 is a timing diagram showing how the TSCs operate.

 FIG. 4 is a flow diagram showing how a master TSC operates.

 FIG. 5 is a flow diagram showing how a slave TSC operates.

25 FIG. 6 shows one example of how the synchronization system may operate in the access network.

DETAILED DESCRIPTION

One technique for synchronizing different line cards in a DOCSIS system is described in copending application Serial No 09/490,761 filed January 24, 2000, entitled: Technique for Synchronizing Multiple Access Controllers at the Head End of An Access Network, which is herein incorporated by reference. The synchronization scheme described below can be used in combination with the synchronizing scheme described this copending application.

A synchronization system synchronizes different CMTSs in different chassis. The synchronization system can also be used to retrofit previous generation CMTSs to achieve internal synchronization between DOCSIS domains within the same chassis.

FIG. 1 shows a DOCSIS system that includes multiple CMTSs 16A-16N that are each connected with an associated optical fiber node 24. Each fiber node 24 is coupled to multiple cable modems 26 that receive IP data over downstream path 20 and transmit IP data over upstream path 22. The CMTS 16A includes a master Timestamp Synchronization Circuit (TSC) 18A and the other CMTSs 16B-16N include slave timestamp synchronization circuits 18B-18N, respectively. In one embodiment, each CMTS 16 is located in a separate chassis. However, other embodiments may have one or more of the CMTSs 16 located in the same chassis.

A central time source 12 generates synchronization pulses 14 and a clock signal 15 to the CMTSs 16A-16N. In one embodiment, the central time source 12 may be located in one of the CMTSs. In another embodiment, the central time source 12 may be a standalone circuit not contained in one of the CMTSs. In one embodiment, each CMTS includes one or more coaxial cable connectors (not shown) that is used to receive the synchronization pulses 14 and the clock signal 15.

5 Through a software protocol, the master TSC 18A asynchronously sends out a message 30 to all the slave TSCs 18B-18N containing a master timestamp value. In one example, the message 30 comprises Internet Protocol (IP) packets sent over a Wide Area Network (WAN) 19. A connection 17 is established by any interface on the CMTS 16A that is capable of sending IP packets over WAN 19. One or more messages 30 contain the IP
10 destination addresses for all CMTSs 16B-16N that need to be synchronized with CMTS 16A. In an alternative embodiment, one message 30 is sent by master TSC 18A that contains a multicast address associated with CMTSs 16B-16N.

In one example, 1 Hertz sync pulses 14 (one pulse per second) or some other low frequency is generated by the central time source 12 and distributed to each CMTS 16. Any
15 frequency for sync pulses 14 can be used that has a long enough period to allow the IP message 30 to be received and processed before the next sync pulse to all slave TSCs 18B-18N. The master timestamp value message 30 is set asynchronously. A local timestamp counter (FIG. 2) in each slave TSC should match the master timestamp value at the occurrence of a next one of the synchronization pulses 14. Any local timestamp value that does not
20 match the master timestamp value is resynchronized with the master timestamp value.

This synchronization scheme allows line cards in different CMTSs to be synchronized. Downstream line cards can be located in separate chassis from upstream line cards. One CMTS can have multiple downstream groups and another CMTS can have multiple upstream groups. For example, an upstream line card 36B in CMTS 16B can
25 process data normally only capable of being processed by an upstream line card 36A in CMTS 16A. Further, a downstream line card 36 in CMTS 16N can send data to cable modems on fiber node 24B that normally could only receive data from CMTS 16B.

5 This provides a synchronization solution for line cards that are not connected together over a common backplane and can not send timing information synchronously between different line cards. This provides more efficient redundancy configurations and allows existing CMTSs to be retrofitted for synchronization in a less backplane pin intensive manner.

10 In the example shown in FIG. 1, the master TSC circuit 18A is located in CMTS 16A. However, the master TSC 18A could alternatively be located in the central time source 12. In this configuration, the central time source 12 has a processor and interface for sending master timestamp values over wide area network 19 and the CMTS 16A has a slave TSC.

Any delay differences from the generation of the sync pulses 14 at the central time
15 source 12 to the arrival at any of the timestamp synchronization circuits 18 will be engineered to be a relative fixed value for all TSCs 18. This could be done by adjusting the master timestamp value according to the difference in delay. Different CMTSs 16 can also allocate portions of that delay between different resources. For example, some CMTSs may have a timestamp bus that would create more delay in the TSC. In other CMTSs, the TSC may
20 receive the sync pulses 14 directly and may not have to add as much or any delay. The TSCs can adjust the received master timestamp value according to these delays.

FIG. 2 shows in more detail the circuitry inside the master TSC 18A and the slave TSC 18B. Each CMTS includes a TSC 18 having a processor or state machine 40 that receives the synchronization pulses 14. The processor or state machine 40 may be the same
25 central processing unit used in the CMTS for processing data or can be a separate circuit just for processing timing signals. The master TSC 18A generates the master timestamp value and sends it in message 30 over the WAN 19 (FIG. 1) or some other portion of an IP

5 network. The slave synchronization circuit 18B receives the master timestamp value 30 from the IP network.

Holding registers 42 store the received master timestamp value for comparing with the value generated by the timestamp counters 44 at the next one of the received sync pulses 14. The timestamp counters 44 are used for counting an amount of time between
10 synchronization pulses 14 and generating a local clock 46. The slave TSC 18B at the next received synchronization pulse 14 compares the master timestamp value stored in holding register 42B with the local timestamp value generated by slave timestamp counter 44B. If the two values match, the slave timestamp counter 44B continues as normal. If the two values do not match, the master timestamp value stored in holding register 42B is loaded into the slave
15 timestamp counter 44B.

FIG. 3 is a timing diagram showing one example of how the slave TSC 18B is synchronized with the master TSC 18A. Referring to FIGS. 2 and 3, the master timestamp counter 44A in the master TSC 18A has a particular timestamp value at pulse 50 of synchronization pulses 14. In this example, the timestamp counter value is thirty. At a next
20 pulse 52, the value of master timestamp counter 44A is thirty five. The processor 40A in master TSC 18A calculates the period T between pulses 50 and 52 to be five counts. The processor 40A predicts that the master timestamp counter 44A will have a value of forty at pulse 54.

After pulse 52, the processor 40A generates the message 30 that identifies the master
25 timestamp value = 40. The message 30 is sent via the wide area network 19 to the CMTS that contains slave TSC 18B. The processor 40B receives the message 30 and stores the master timestamp value=40 in holding register 42B. At the next received sync pulse 54, the

5 processor 40B compares the value in holding register 42B with the local timestamp value output by slave timestamp counter 44B.

If the two values match, or come within some predetermined range, the slave timestamp counter 44B continues counting with no reload. If the master timestamp value (MTV) in holding register 42B and the output of slave timestamp counter 44B at pulse 54 do
10 not match, or come within the predefined range, then the slave timestamp counter 44B is loaded with the master timestamp value 40 in holding register 42B.

In another embodiment, the master TSC 18A may predict the master timestamp value for some other pulse, other than the immediately following pulse 54. For example, the master TSC may predict the master timestamp value for two clock pulses after pulse 52. This would
15 allow more time for the master timestamp value to arrive and be processed by the slave TSCs. In another embodiment, the sync pulse used as a reference for comparing to the master timestamp value is marked to distinguish it over other sync pulses.

FIG. 4 shows the operations performed for the master TSC 18A shown in FIGS. 1 and 2. In block 62, the master TSC receives the sync pulses 14. The previous calculation for the
20 expected master timestamp value at a next sync pulse is compared with the actual value output from the master timestamp counter 44A (FIG. 2). If the two values are not the same in decision block 64, an error routine is executed in block 82. This may include sending an error message to one of the slave synchronization circuits. The circuit receiving the error message could then take over operation as a new master TSC.

25 If the two values are the same, or within some predetermined range, a wait count is set in block 68. The wait count is set when a master timestamp value is not sent for each sync pulse. This may be done to reduce system data traffic. The master waits in block 70 until

5 another sync pulse is received in block 72 and then decrements the wait count in block 74.
The master receives sync pulses until the wait count drops to zero in decision block 76.

The master timestamp value is predicted in block 78. For example, as previously described in FIG. 3, the number of counts that are expected to occur for a period T between adjacent sync pulses is determined. The processor 40A (FIG. 2) then adds the count value of
10 the master timestamp counter 44A (FIG. 2) at the last received sync pulse with the number of counts previously calculated for a period T between adjacent sync pulses. This value is referred to as the master timestamp value. The master TSC in block 80 then sends the master timestamp value in a IP message to the slave TSCs located in other CMTSS.

FIG. 5 shows the operations performed by one of the slave TSCs, such as slave TSC
15 18B or 18N. The slave TSC receives the master timestamp value via an internet connection or some other sort of asynchronous data transfer in block 106. The master timestamp value is loaded into a local holding register 42B (FIG. 2) in block 108. The slave TSC receives a next synchronization pulse after receiving the master timestamp value in block 90. An error load count value (Error_Load_Count) is decremented in block 92. In block 94, the master
20 timestamp value previously stored in the local holding register is compared with the local timestamp value generated by the slave timestamp counter 44B (FIG. 2).

If the two values are the same, or within some predefined acceptable range in decision block 96, the slave TSC returns to wait for a next master timestamp value in block 106. If the two timestamp values are not the same, or not within the predefined range, then the slave
25 timestamp counter 44B is loaded with the master timestamp value previously stored in the holding register 42B in block 98.

Blocks 92, 100, 102 and 104 provide an error checking routine. The Error_Load_Count value tracks how many times a slave TSC is reloaded with the master

5 timestamp value for some ratio of received synchronization pulses. For example, every time the slave timestamp counter is reloaded, the Error_Load_Count is increased by some value"x". The Error_Load_Count is then decremented in block 92 for every synchronization pulse received in block 92. If the Error_Load_Count exceeds some predetermined threshold in decision block 102, an error routine is executed in block 104. The error routine may log an
10 error message, send an error message to a system administrator, or may automatically switch in a redundant line card.

In order to illustrate how the technique of the present invention may be used to overcome some of the limitations associated with conventional cable network configurations, an example of a video-on-demand application will now be described using the access
15 network shown in FIG. 6. The TSCs 18A and 18B for CMTS 16A and 16B, respectively, are included at the head end of the cable network and may include hardware and/or software which is used to synchronize selected line cards within the CMTSs.

The TSCs 18A and 18B update current time reference data, and periodically distribute local clock signals 46A and 46B to each (or a selected portion) of Media Access Control
20 (MAC) 106 and 108, respectively. The TSCs 18A and 18B may alternatively reside inside the MACs 106 and 108, respectively. By synchronizing each of the TSCs as described above, each MAC within each CMTS may be synchronized with other MACs within other CMTSs, thereby resulting in each line card in each CMTS being in synchronization.

In FIG. 6, a downstream channel A (113) and downstream channel B (123) are RF
25 combined and connected to a single optical fiber which carries the downstream signals to both optical node 152A and optical node 152B. Thus, each of the cable modems within group 160A and group 160B are able to receive both downstream channel 113 and downstream channel 123.

5 In this example, it is assumed that each downstream channel A (113) and B (123) is provided sufficient bandwidth for simultaneously broadcasting a plurality of different movies or other video data. Further, it is assumed that a user connected to cable modem CM1 (161) has previously been watching a movie on downstream channel A and communicates with the CMTS 16A via upstream channel A1 of upstream path 127. In this example, the user at CM1
10 now wishes to watch a movie which will be broadcast on downstream channel B.

At this point, the CMTSs have a number of different options by which to proceed. First, the CMTS 16A may provide the desired movie to CM1 on downstream channel A. However, even assuming that the cable operator has the additional bandwidth to provide this movie on downstream channel A, this option is undesirable as it is considered to be a waste of
15 resources to broadcast the identical movie on two different downstream channels.

Alternatively, a preferred solution would be for the CMTS 16A to instruct the cable modem CM1 to switch downstream channels and receive the movie on downstream channel B.

In conventional cable networks, this option would not be available since, without synchronization between the two line cards A (103) and B (104), it would not be possible for
20 the cable modem CM1 to "listen" to the CMTS 16B on downstream channel B and "talk" to the CMTS 16A on upstream channel A1. However, using the synchronization technique of the present invention, the cable modem CM1 is able to obtain current timestamp data from downstream channel B associated with line card B in CMTS 16B, and use the timestamp data to synchronize itself with line card A in order to "talk" to the CMTS 16A via upstream
25 channel A1. The TSCs 18A and 18B allow each respective MAC controller 106 and 108 to be in synchronization. Accordingly, cable modem CM1 (161) is able to use the timestamp message on downstream channel B (123) to communicate with the upstream receivers 105 on line card A.

5 Referring to the video-on-demand example described above, when the cable modem CM1 sends a request to the CMTS 16A to view a movie which is currently being broadcast on downstream channel B, the CMTS 16A may respond by instructing the cable modem CM1 to switch its downstream channel from downstream channel A to downstream channel B. The cable modem CM1 is then able to “listen” to the CMTS 16B on downstream channel B, and “talk” to the CMTS 16B using any one of the upstream A channels 119A or 119B.

Cable modems in group 160B can receive data from downstream channel A or downstream channel B. Cable modems in group 160B send data over upstream channels 129. A modification may be made whereby the upstream and downstream ports on each line card are connected to both optical node 152A and optical node 152B. In this modified
15 embodiment, each of the cable modems in the network has access to the ports on both line card A and line card B.

Initially, it may be assumed that line card A in CMTS 16A services the cable modems of group 160A, and line card B in CMTS 16B services the cable modems of group 160B. In accordance with the technique of the present invention, if a problem occurs on line card A,
20 for example, the group 160A modems are able to switch over to line card B in CMTS 16B without these modems having to resynchronize themselves with the line card B time reference (since line card A is already synchronized with line card B).

In conventional systems, the two line cards in separate CMTSs would not be synchronized. Thus, any modems switching from line card A to line card B are required to
25 re-synchronize with line card B. This introduces delays in the communication protocol between the cable modem and the CMTS. In certain applications, such as telephony, such delays are extremely undesirable since they directly effect the call quality of a voice call.

5 Thus, the technique of the present invention may be used to synchronize a plurality of different access controllers which control a plurality of distinct ports at the Head End of an access network. In the context of a cable network, the technique of the present invention may be used to synchronize desired upstream and/or downstream channels across different line cards within different Cable Modem Termination Systems (CMTS).

10 The synchronization scheme described in Serial No 09/490,761 filed January 24, 2000, can be used for distributing a synchronized local timestamp for a particular CMTS to different line cards connected together on the same CMTS backplane.

 The synchronization techniques described above offer a number of distinct advantages over conventional techniques used in the configuration or design of access networks. For
15 example, there are a limited number of line cards slots in each CMTS chassis. In current CMTS architectures, both upstream and downstream line cards must be located in the same chassis. That means one upstream and one downstream redundant line card is required for each chassis (3+1 and 3+1).

 The present invention allows downstream line cards to be located in the same chassis
20 and upstream line cards to be located in the same chassis. Then only one redundant line card is required for each chassis. For example, if a chassis was limited to eight line card slots, then seven slots could be used for downstream line cards, and only one slot would be required for a redundant downstream line card (7+1). Another CMTS is used for seven upstream line cards and one redundant upstream line card (7 +1). Thus, the synchronization scheme
25 described above saves one line card slot for each eight slot chassis.

 In addition to providing benefits for redundancy protocols, the timestamp synchronization technique of the present invention provides for seamless downstream channel change at the cable modem end. Timestamp synchronization also provides benefits in

5 facilitating multi-service convergence of voice, video, and high-speed data applications. These issues become increasingly important as streaming media and video streams are multiplexed onto the same data network.

Additionally, the technique of the present invention provides added flexibility in network implementation by allowing DOCSIS (or MAC) domains to be dynamically
10 configurable via software. Further, each DOCSIS domain may be configured to cross line card and CMTS boundaries. Thus, the technique of the present invention provides the advantage of allowing different upstream and/or downstream ports on different line cards to be grouped together within the same DOCSIS domain.

This provides the advantage of allowing greater flexibility in the design of line card
15 interfaces. Furthermore, since different ports on different line card interfaces may be assigned to the same domain, the cable operator or service provider is allowed greater flexibility and scalability in configuring different domains to suit the needs specific applications such as, for example, telephony, video-on-demand, etc.

The system described above can use dedicated processor systems, micro controllers,
20 programmable logic devices, or microprocessors that perform some or all of the operations. Some of the operations described above may be implemented in software and other operations may be implemented in hardware.

For the sake of convenience, the operations are described as various interconnected functional blocks or distinct software modules. This is not necessary, however, and there
25 may be cases where these functional blocks or modules are equivalently aggregated into a single logic device, program or operation with unclear boundaries. In any event, the functional blocks and software modules or features of the flexible interface can be

5 implemented by themselves, or in combination with other operations in either hardware or software.

Having described and illustrated the principles of the invention in a preferred embodiment thereof, it should be apparent that the invention may be modified in arrangement and detail without departing from such principles. Claim is made to all modifications and
10 variation coming within the spirit and scope of the following claims.